2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
17-20 October, 2021. Melbourne, Australia

Skill-level-based Hybrid Shared Control for Human-Automation Systems*

Sooyung Byeon[1], Dawei Sun[1], and Inseok Hwang[1]

2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-4207-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMC52423.2021.9658994

***Abstract*— In this paper, a hybrid shared controller is proposed for assisting human novice users to emulate human expert users within a human-automation interaction framework. This work is motivated to let human novice users learn the skills of human expert users using automation as a medium. Automation interacts with human users in two folds: it learns how to optimally control the system from the experts demonstrations by offline computation, and assists the novice in real time without excess amount of intervention based on the inference of the novice's skill-level within our properly designed shared controller. Automation takes more control authority when the novices skill-level is poor, or it allows the novice to have more control authority when his/her skill-level is close to that of the expert to let the novice learn from his/her own control experience. The proposed scheme is shown to be able to improve the system performance while minimizing the intervention from the automation, which is demonstrated via an illustrative human-in-the-loop application example.**

I. INTRODUCTION

Human-automation interaction has been employed in the fields of driver assistance systems [1], robots [2], teleoperation systems [3], and exoskeletons [4], so that human users can perform complex tasks through collaboration between the human and automation. In particular, the shared control frameworks have recently been studied to control the system safely and effectively by the automation assisting the human when the human user lacks the skill-level for controlling the system [5]. Along with many applications, the shared control has many interpretations or definitions [6]. In this paper, we refer the shared control as a control scheme where the human user and automation share the same control space, and the control input of the system is the weighted average (or blending) of human input and automation input. That weighting is called *control authority* [7], [8]. Various approaches such as game theoretic methods [9], optimal control [1], and reinforcement learning [5] have been used to dynamically allocate control authority depending on scenarios.

Many existing works have established various shared control schemes for driving [10]–[13], robot control [14], and specific missions [15]–[17]. Each related work presents methods to determine the control authority according to the situation. A large number of schemes introduce automation to the shared control only when the system is about to violate predefined safety conditions [15] or minimize the intervention from the automation [5]. Automation provides awareness of the situation, assesses the performance in real time, and allocates sufficient amount of control authority to the human user whenever the human input does not deteriorate the system's performance significantly. As human input is accepted in the shared control systems, human users can experience controlling the system to improve their skill-levels without loss of situational awareness and avoiding rejection of the assistance [18], [19]. Existing frameworks are well designed considering human factor, but there are two limitations. First, to assure that the automation has the correct mission objective to the optimization problem which means that for a certain mission, the shared control system should follow a predefined control criterion regardless of the user's perception or understating of the system [20]. However, a correct mission objective might not be easily predefined depending on the task. Second, existing interactive schemes typically do not take into account the different characteristics of human users. In practice, there are not only differences between experts and novices, but also differences between novices. A concept of skill-based personalized assistance was introduced in [21], but it does not provide a shared control.

A skill-level-based hybrid shared control framework is proposed to overcome the two major limitations of the existing frameworks by learning the mission objective from a human expert user and customizing the shared control scheme based on a skill-level of a human novice user. We define the skill-level as a value indicating the degree to which a human user adheres to the expert's mission objective. The expert's mission objective can be inferred from demonstrations of the expert by using a data-driven method, such as inverse optimal control (IOC) [22]. The inferred objective is employed as a performance measure to evaluate the time-varying skill-level of human novice users using another data-driven method, such as online dynamic mode decomposition (DMD) [23]. Then, the control authority is allocated in real time: more control authority to the novice when his performance is close to that of the expert, or more control authority to the automation when the novice's performance is poor. The control authority is assigned in a discrete manner by determining the *discrete mode* of the hybrid systems [24]. Human users can perceive discrete control authority better than continuous values and the number of discrete modes can be adjusted depending on the scenarios.

The rest of the paper is organized as follows. In Section II, we propose the hybrid shared control scheme. The hybrid shared controller is formally presented in Section III. In Section IV, the proposed hybrid shared controller is tested and demonstrated via an illustrative example using a quadrotor simulator. Conclusions are given in Section V.

*The authors would like to acknowledge that this work is supported by NSF CNS-1836952.

[1]School of Aeronautics and Astronautics, Purdue University, IN, 47907 USA {sbyeon,sun289,ihwang}@purdue.edu

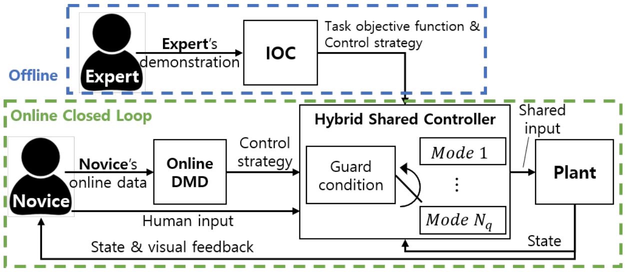

Fig. 1. Skill-level-based hybrid shared control scheme.

II. Skill-Level-Based Hybrid Shared Control

We propose a skill-level-based hybrid shared control scheme which is designed for assisting a human novice user in a personalized way based on his skill-level to emulate a human expert user with minimum intervention from an automation (Fig. 1). The proposed scheme employs the expert as a data source such that the automation can extract a mission objective and control strategy from his demonstration by offline computation. The novice is modeled and assisted online by a hybrid shared controller which allocates the control authority based on the skill-level discrepancy between the expert and the novice while avoiding excessive assistance. In this section, we introduce and design each element of the proposed scheme: a plant, a human expert user, a human novice user, and a hybrid shared controller.

A. Plant

We consider a linear time-invariant (LTI) plant. Note that many practical systems can be modeled as LTI plants [25].

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{1}$$

where $\mathbf{x}(k) \in \mathcal{X} \subseteq \mathbb{R}^n$, $\mathbf{u}(k) \in \mathcal{U} \subseteq \mathbb{R}^m$, and $\mathbf{x}_0 \in \mathbb{R}^n$ denote the continuous state, shared input, and initial state, respectively. $k \in \mathcal{T} = \{0, 1, \cdots, T\}$ is the time index. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ denote the system matrices. We assume that the system (A, B) is controllable so that the problem is well-posed.

B. Expert Performance Modeling

An expert provides his demonstrations which are used to extract a quantified mission objective and his control strategy. We consider the expert to be a human user who is capable of optimally controlling the plant based on his knowledge and experience. Thus, the expert's behaviors can be interpreted as an optimal control strategy with an unknown mission objective. Note that when the expert operates the system, there is no intervention from the automation.

We formulate the expert's task objective function $J_e(D)$ as a quadratic function [14], with the expert's demonstration $D_e = \{\mathbf{x}_e, \mathbf{u}_e\}^{[0,T]}$ which denotes the time history of the state and control input from the expert for time step $k \in \mathcal{T}$:

$$D_e = \arg\min_D J_e(D) \tag{2}$$

subject to (1) and

$$J_e(D) = \sum_{k=0}^{T} \left(\mathbf{x}(k)^T Q \mathbf{x}(k) + \mathbf{u}(k)^T R \mathbf{u}(k)\right) \tag{3}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ denote the unknown task objective matrices which are positive definite.

IOC is used to quantify the task objective matrices (Q, R) in (3) from the given expert's demonstration D_e [22]. An expert's task objective function can be described by the feature $\Phi(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^r$:

$$J_e(\mathbf{w}, D_e) = \sum_{k=0}^{T} \Phi(\mathbf{x}_e(k), \mathbf{u}_e(k)) \cdot \mathbf{w} \tag{4}$$

where $\mathbf{w} \in \mathbb{R}^r$ is the feature weight and the elements of feature $\Phi = [\phi_1, \phi_2, \cdots, \phi_r]^T$ are quadratic forms, such as $\phi_i = x_i^2$ or $\phi_j = u_j^2$. Using IOC, we can compute the weight $\mathbf{w}$, or equivalently (Q, R), from the given demonstration D_e and the system model (1). Suppose $\{\mathbf{x}_e, \mathbf{u}_e\}^{[t:t+l-1]}$ denotes a piece of demonstration with the observation length l. The Karush-Kuhn-Tucker (KKT) condition implies:

$$\mathbf{H}^{t,l} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\lambda}(t+l) \end{bmatrix} = 0 \tag{5}$$

where $\mathbf{H}^{t,l}$ denotes the recovery matrix which can be obtained using the plant dynamics (1), state, and control input. $\boldsymbol{\lambda} \in \mathbb{R}^n$ denotes the dual variable. If the observation length l is long enough and the demonstration D_e is informative enough in a distinguishable manner, then $\mathbf{w}$ can be uniquely determined with a normalization ($\|\mathbf{w}\| = 1$) [22].

An expert control strategy can be equivalently represented as a linear state feedback controller by a linear-quadratic regulator scheme using the linear system dynamics (1) and the quadratic task objective function (4):

$$\mathbf{u}_e(k) = K_e \mathbf{x}(k), \quad \forall k \in \mathcal{T} \tag{6}$$

where $K_e \in \mathbb{R}^{m\times n}$ denotes the expert control gain and $\mathbf{u}_e(k) \in \mathcal{U}$ denotes the modeled expert input. Note that the system dynamics (1) with the expert control strategy (6) is always stable, i.e., $\max(|\lambda(A+BK_e)|) < 1$ where λ denotes the eigenvalue, which is reasonable since we assume that the expert know how to operate the system optimally.

C. Novice Performance Modeling

A novice is a target user of the proposed scheme whose skill-level is assessed by the automation based on his control strategy. Unlike the expert case, we do not assume that a novice control strategy optimizes anything. For instance, the novice may not be able to operate the system properly due to his lack of skill and experience. Another issue of the novice modeling is variability of the novice's skill-level: the novice's skill-level may change while performing a task. We model the novice control strategy as a feedback controller but use a linear approximation:

$$\mathbf{u}_n(k) = K_n(\mathbf{x}(k), k) \approx \hat{K}_n(k)\mathbf{x}(k), \quad \forall k \in \mathcal{T} \tag{7}$$

where $K_n(\cdot)$ denotes the true but unknown novice's feedback control strategy, $\mathbf{u}_n(k) \in \mathcal{U}$ is the modeled novice input, and $\hat{K}_n(k) \in \mathcal{K}_n \subseteq \mathbb{R}^{m\times n}$ denotes the linear approximation of the novice model.

We use the online DMD to obtain and update the novice control strategy $\hat{K}_n(k)$ in real time using the novice input and state of the plant. The online DMD is known to provide efficient and accurate representation of the time-varying system and widely accepted in control system identification [23]. Let $\mathbf{y}(k)$ be the resultant state with the current state $\mathbf{x}(k)$ and the novice input $\mathbf{u}_n(k)$ at time step k without assistance, i.e., $\mathbf{y}(k) = A\mathbf{x}(k) + B\mathbf{u}_n(k)$. The goal of the online DMD is to find a matrix $\tilde{A}(k) = A + B\hat{K}_n(k)$. To obtain $\tilde{A}(k)$, the following cost function is minimized:

$$\tilde{J}(k) = \sum_{i=1}^{k} \|\sigma^{k-i}\mathbf{y}(i) - \sigma^{k-i}\tilde{A}(k)\mathbf{x}(i)\|^2 \tag{8}$$

where $\sigma \in (0, 1]$ denotes the discount factor. The given pairs of snapshots for $i = 1, 2, \cdots, k$ form matrices:

$$\begin{aligned} \mathbf{X}(k) &= [\sigma^{k-1}\mathbf{x}(1) \quad \sigma^{k-2}\mathbf{x}(2) \quad \cdots \quad \mathbf{x}(k)] \\ \mathbf{Y}(k) &= [\sigma^{k-1}\mathbf{y}(1) \quad \sigma^{k-2}\mathbf{y}(2) \quad \cdots \quad \mathbf{y}(k)] \end{aligned} \tag{9}$$

where both have the same dimension $\mathbb{R}^{n\times k}$. Note that we consider the over-constrained problem, i.e., $k > n$. The least-square solution gives the following result.

$$\tilde{A}(k) = \mathbf{Y}(k)\mathbf{X}(k)^\dagger = \mathbf{Y}(k)\mathbf{X}(k)^T \left(\mathbf{X}(k)\mathbf{X}(k)^T\right)^{-1} \tag{10}$$

where † denotes the MoorePenrose pseudoinverse. In the next time step $k+1$, $\tilde{A}(k+1)$ can be obtained as a recursive least-square form [23].

$$\tilde{A}(k+1) = f_{\text{DMD}}\left(\tilde{A}(k), \mathbf{y}(k+1), \mathbf{x}(k+1), \sigma\right) \tag{11}$$

where f_{DMD} denotes the recursive online DMD operation. Finally, $\hat{K}_n(k)$ can be computed as follows:

$$\hat{K}_n(k) = B^\dagger(\tilde{A}(k) - A) \tag{12}$$

Note that we assume $\hat{K}_n(k)$ is slowly varying in time [23] which means the novice's skill-level is not rapidly changing.

D. Hybrid Shared Controller Design

The hybrid shared controller (Fig. 1) allocates the optimal control authority and provides the shared input with respect to the novice's skill-level. The control authority is discretized and maps into the discrete mode of the hybrid system. The automation input is generated based on the expert control strategy. Thus, the hybrid shared controller performs four functions in real time: i) generating automation input which mimics the expert control strategy (6), ii) assessing the novice's skill-level and control strategy (7), iii) determining the discrete mode and allocating the control authority based on the novice's skill-level in terms of trading-off between the performance of the system and reduced amount of intervention, and iv) generating the shared input using novice input, automation input, and determined control authority. Note that i) and ii) are already presented in Sections II-B and II-C, respectively. Thus, we explain iii), the control authority allocation method which is called a *guard condition* and iv), how to produce the shared input in Section III.

III. Hybrid Shared Controller

The hybrid shared controller generates the shared input as a final output to improve the performance of a human-in-the-loop system with a minimum amount of intervention, as described in Section II-D. In this section, we explain how to determine the optimal control authority and generate the shared input with respect to the novice's skill-level. The hybrid shared controller can be formally written as a tuple of elements:

$$\mathcal{H} = (\mathcal{Q}, \mathcal{I}, \mathcal{O}, f, \mathcal{G}) \tag{13}$$

where $\mathcal{Q}$ is a finite set of the discrete mode, $\mathcal{I}$ is the input, $\mathcal{O}$ is the output, f stands for the shared control law, which addresses how the shared input is generated under each mode, and $\mathcal{G}$ denotes the guard condition that assigns the previous mode to the current mode under a mode switching condition. Each element of the tuple is designed as follows to allocate the optimal control authority:

- $\mathcal{Q} = \{1, 2, \cdots, N_q\}$, i.e., the hybrid shared controller has total N_q modes. With larger N_q, the control authority is discretized more finely. However, if N_q is too large, it might be difficult for novices to recognize the change in the control authority, and the amount of computation increases.
- $\mathcal{I} = \mathcal{X} \times \mathcal{U} \times \mathcal{K}_n$, i.e., inputs of the hybrid shared controller are the state, human (novice) input, and novice control strategy.
- $\mathcal{O} = \mathcal{U}$, i.e., output of the hybrid shared controller is the shared input $\mathbf{u}(k)$.
- $f : \mathcal{X} \times \mathcal{U} \times \mathcal{Q} \to \mathcal{O}$ is the shared control law.

$$\begin{aligned} \mathbf{u}(k) &= f(\mathbf{x}(k), \mathbf{u}_h(k), q(k)) \\ &\triangleq \theta(q(k))\mathbf{u}_h(k) + (1 - \theta(q(k)))\mathbf{u}_a(k) \end{aligned} \tag{14}$$

where $\theta(q(k)) : \mathcal{Q} \to [0, 1]$ is the control authority, $q(k) \in \mathcal{Q}$ denotes the mode, $\mathbf{u}_h(k)$ is the human input, and $\mathbf{u}_a(k)$ is the automation input which is designed as:

$$\mathbf{u}_a(k) = K_e\mathbf{x}(k) \tag{15}$$

In words, f is the weighted average of human input and automation input. When $\theta = 0$, the automation takes full control authority, and when $\theta = 1$, the human user has full control authority. Any value between 0 and 1 can be set as a discrete control authority depending on the scenarios.

A core part of the hybrid shared controller is to design the guard condition which determines a mode transition, or changing the control authority in the proposed scheme. We define a loss function L, which is a tool for designing the guard condition. The loss function is an estimated future performance with (1), (7), (15), and the hybrid shared controller (13). To estimate the future performance, it is assumed that the novice model (7) at time step t is the valid estimate for the future time horizon $[t, t+N]$ which can be interpreted that the future behavior of the human novice user over time $[t, t+N]$ is predicted based on the current novice model at time step t. Note that it is reasonable since we assume that the novice model is slowly time-varying. Then, the loss function $L : \mathcal{X} \times \mathcal{K}_n \times \mathcal{Q} \rightarrow \mathbb{R}^+$ is defined as:

$$
\begin{aligned}
L_{\theta(q(t))}^{[t,t+N]} \triangleq & \sum_{k=t}^{t+N} \big(\mathbf{x}(k)^T Q \mathbf{x}(k) + \hat{\mathbf{u}}(k;t)^T R \hat{\mathbf{u}}(k;t) \\
& + \alpha(1-\theta(q(t)))^2 \mathbf{u}_a(k)^T R \mathbf{u}_a(k)\big) \\
= & \sum_{k=t}^{t+N} \big(\mathbf{x}(k)^T (Q + K_s(t)^T R K_s(t) \\
& + \alpha(1-\theta(q(t)))^2 K_e^T R K_e\big)\, \mathbf{x}(k)\big)
\end{aligned}
\tag{16}
$$

where

$$
\begin{aligned}
\hat{\mathbf{u}}(k;t) &\triangleq K_s(t)\mathbf{x}(k), \ \forall k \in [t, t+N] \\
K_s(t) &= \theta(q(t))\hat{K}_n(t) + (1-\theta(q(t)))K_e
\end{aligned}
\tag{17}
$$

and $\hat{\mathbf{u}}(k;t)$ denotes the best estimate of the shared input over time $k \in [t, t+N]$ at time step t. $K_s(t)$ denotes the shared control gain at time step t, $\alpha > 0$ denotes the tunable penalty parameter on the assistance from the automation, and N is a finite horizon. The loss function can be broken down into two parts: the first part has the same structure as a task objective function of the expert (3) and the second part denotes a penalty on the assistance from the automation; If $\alpha = 0$, then the loss function is always minimized with $\theta = 0$, which means the automation takes the full control authority all the time. A positive penalty α ensures that the hybrid shared controller provides assistance only when the novice's skill-level is different from that of the expert. Based on the loss function, the guard condition is designed as follows:

- $\mathcal{G} : \mathcal{Q} \times \mathcal{Q} \rightarrow 2^{\mathcal{X} \times \mathcal{K}_n}$, i.e., the guard condition assigns the previous discrete mode $i = q(k-1) \in \mathcal{Q}$ to the current discrete mode $j = q(k) \in \mathcal{Q}$ if the state and the novice control strategy are inside of the guard condition.

$$
\begin{aligned}
&\mathcal{G}(i,j) = \{\mathbf{x}(k), \hat{K}_n(k) \mid \mathbf{x}(k) \in \mathcal{X}, \hat{K}_n(k) \in \mathcal{K}_n, \\
&j \in \mathcal{Q}, l \in \mathcal{Q}, j \neq l, L_{\theta(j)}^{[k,k+N]} \leq L_{\theta(l)}^{[k,k+N]}\}
\end{aligned}
\tag{18}
$$

Note that the proposed guard condition implies:

$$
q(k) = \arg\min_{q(k)} L_{\theta(q(k))}^{[k,k+N]} \tag{19}
$$

Note that the upper bound of the loss function is given when $\theta = 0$ (i.e., the automation has full control authority) with a stable expert control strategy. The guard condition with that upper bound ensures that the performance of the system is also bounded regardless of the novice's skill-level.

IV. Application Example

We present an application example with human subjects to demonstrate how the proposed scheme works and compare with a baseline method.

A. Test-bed

We design an experimental test-bed, called the quadrotor landing simulator (Fig. 2). The human users are requested to land a quadrotor slowly on the touch-pad using a joystick while checking the current status via a monitor. The quadrotor dynamics is linearized with respect to an equilibrium point [26]. The continuous states consist of the position, velocity, attitude, and angular velocity; $\mathbf{x} = [x, y, \dot{x}, \dot{y}, \psi, \dot{\psi}]^T$. The time interval for discretization $\Delta t = 0.033$ seconds and the state domain $\mathcal{X} = [-512, 512] \times [0, 768]$ in pixels are set, respectively. The shared input is set to $[u_x, u_y]^T \in [-1, 1]^2$. An initial position is randomly generated but uniformly distributed in $x(0) \in [-412, 412]$ and $y(0) \in [568, 668]$, respectively. All other states are equal to zero initially.

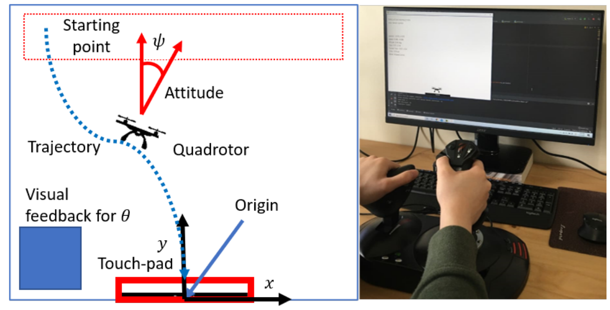

Fig. 2. (Left) Quadrotor landing simulator: a screen shown in the simulator. (Right) Test-bed configuration. The current control authority is indicated at the left-lower corner: {blue, yellow, red} = $\{1, 0.5, 0\}$.

To demonstrate the efficacy of the proposed hybrid shared controller, we consider three discrete modes for control authority allocation: $\{\theta(1), \theta(2), \theta(3)\} = \{1, 0.5, 0\}$. These three modes denote the full control authority to the human user, half-control authority to the human user, and full control authority to the automation, respectively. The initial discrete mode for about 1.7 seconds is $q = 1$ as the online DMD requires minimum time for modeling the novice user from his real time demonstrations. Other parameters are set as follows; the penalty parameter $\alpha = 10$, the discount factor $\sigma = 0.9$ (refer to [23] for a detailed method to determine a proper discount factor), and the finite time horizon $N = 200$ (about 7 seconds in real time). A basis function for IOC is set to the quadratic terms of the states and control inputs; $\mathbf{\Phi} = [x^2, y^2, \dot{x}^2, \dot{y}^2, \psi^2, \dot{\psi}^2, u_x^2, u_y^2]^T$. Then, the feature weight $\mathbf{w}$ denotes the diagonal terms of (Q, R) matrices in (3), i.e., $Q = \text{diag}(w_1, w_2, w_3, w_4, w_5, w_6)$ and $R = \text{diag}(w_7, w_8)$.

B. Human Subjects and Procedure

We first model an expert who is experienced with the simulator. The primary objective for the expert is to land the quadrotor slowly without crashing. Total 10 demonstrations are provided by the expert and the inferred weight vector $\mathbf{w}$ in (4) is obtained by an average value. Novices, who are new to the simulator, are requested to land the quadrotor slowly with or without the hybrid shared controller. An exercise session is given before recording their performance. A total of 6 novices participated in the experiment.

C. Results and Discussion

1) Individual Cases: We first present two individual cases to show how the proposed scheme works in practice.

- *Case 1* corresponds to a result of a novice with relatively poor skill-level.
- *Case 2* represents another result of a novice who performs better than *Case 1* but worse than the expert.

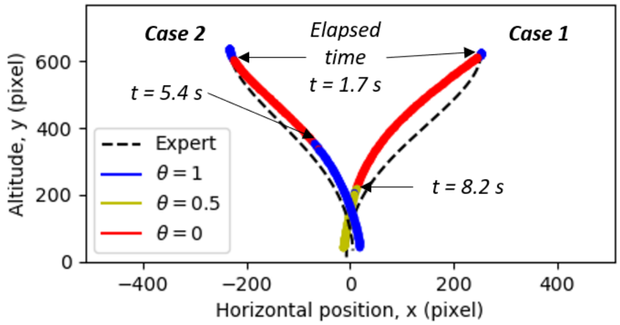

Fig. 3. Two individual trajectories of a human-in-the-loop system with the hybrid shared controller: *Case 1* and *Case 2* (Individual novice 1 with poor skill-level and individual novice 2 with better skill-level).

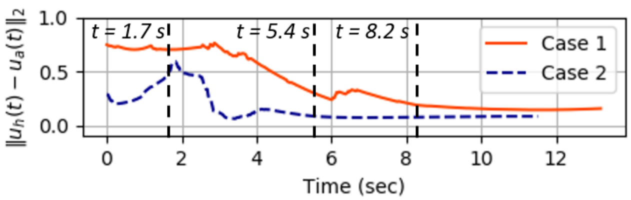

Fig. 4. Time history of control input discrepancy: *Case 1* and *Case 2*.

In *Case 1*, the novice's trajectory deviates rather largely from that of the recovered expert (Fig. 3). Note the expert's trajectory is computed using the inferred expert control strategy (6). The hybrid shared controller begins to assist the novice immediately after the first online DMD modeling is finished at about 1.7 seconds by giving the full control authority to the automation, and it is maintained until about 8.2 seconds. Even after 8.2 seconds, the automation shall not pass over full control authority to the novice, but only half of it, which shows that the novice's skill-level is poor and how the proposed scheme deals with it.

In *Case 2*, the novice's trajectory deviates less from that of the expert. The control authority is taken by the automation after initial 1.7 seconds, same as *Case 1*, but it is observed that the control input discrepancy, which is defined as $\|\mathbf{u}_h(t) - \mathbf{u}_a(t)\|_2$, deceases after 1.7 seconds (Fig. 4), which reveals that the inferred skill-level of the novice is improving over time. As a result, the novice takes over the full control authority again and the control input discrepancy is maintained at a significantly smaller level than *Case 1*. This means that the novice shows a similar skill-level to the expert so that the proposed scheme allocates more control authority to the novice over time.

2) Comparison with Baseline: A baseline method is examined along with the proposed method for comparison. We choose the Maxwell's demon algorithm (MDA) [14] as a baseline method. The MDA has only two discrete modes; one is fully accepting human user input when the human input and the automation input are in the same half-plane. Otherwise, another mode completely ignores the human user input and provides zero shared input.

Each human novice user is requested to perform the same quadrotor landing task. After they have an exercise

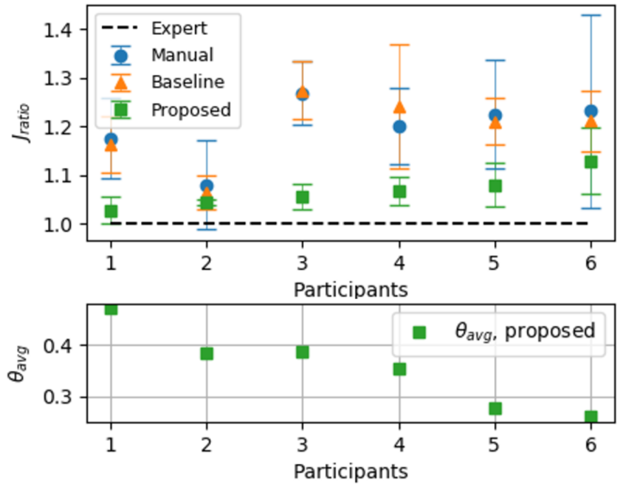

Fig. 5. (Top) Control performance comparison between manual control, baseline (MDA), and proposed (hybrid shared controller) based on J_{ratio}. Error bars denote the standard deviation. (Bottom) Averaged control authority of the proposed scheme is used. More control authority is given to each participant when his performance is close to that of the expert.

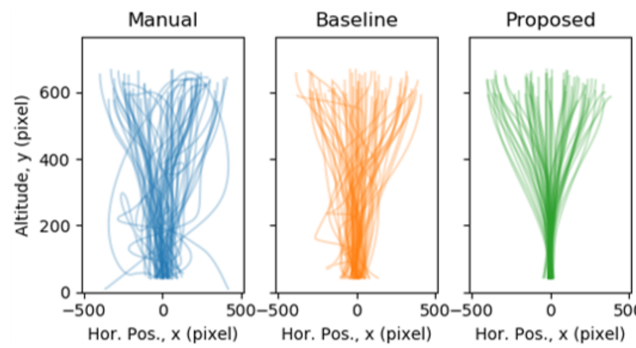

Fig. 6. All recorded trajectories for three different control schemes (6 participants, $n = 60$ for each scheme): manual control, baseline (MDA), and the proposed hybrid shared control scheme.

session, they perform the task with three different schemes 10 times for each: i) manual control without assistance, ii) assisted by the baseline, and iii) assisted by the proposed scheme. In order to compensate for the difference which might be caused by the order of using each method, the order of conducting each experiment is randomized [14]. The following performance measures are used.

- The expert's task objective function (4) is used to compare the performance. Since the value of the task objective function depends on the initial state, we measure the ratio of the evaluated task objective function for each trial: $J_{\text{ratio}} \triangleq J_e(D_n)/J_e(D_e)$ (closer to 1 is better) where D_e and D_n are demonstrations from an expert and a novice, respectively.
- The averaged control authority, which is defined as $\theta_{\text{avg}} \triangleq 1/T \sum_{k=0}^{T} \theta(k)$, is shown to validate that the proposed scheme indeed allows more control authority to the novice when they perform close to the expert, thereby reducing interference from the automation when unnecessary.

In Fig. 5, the performance measure J_{ratio} for all participants is presented. The order of the participants is sorted in ascending order based on the J_{ratio} with the proposed scheme (green points). Manual control and baseline have no statistically significant difference in J_{ratio} (one-way ANOVA $p > 0.5$) and the proposed scheme does improve J_{ratio} compared to the manual control ($p < 0.001$). In Fig. 5, when a novice has better skill-level (lower J_{ratio}), more control authority is given to him as we design. Note that the averaged control authority θ_{avg} can be adjusted by tuning the penalty parameter α: If $\alpha = 0$, the performance should be same as the expert ($J_{\text{ratio}} = 1$) only with automation input, i.e., $\mathbf{u}(k) = \mathbf{u}_a(k)$ and $\theta_{\text{avg}} = 0$. If $\alpha \to \infty$, then the hybrid shared control scheme is equivalent to the manual control, i.e., $\mathbf{u}(k) = \mathbf{u}_h(k)$ and $\theta_{\text{avg}} = 1$.

Finally, all trajectories are presented in Fig. 6. It is shown that the proposed scheme can also improve the consistency of trajectories, i.e., the landing performance is more consistent across the novice users. This shows that the proposed scheme is able to assist novices to emulate the expert, as we design.

V. CONCLUSIONS

A hybrid shared control algorithm that can explicitly account for the human user's skill-level was proposed to improve the performance of human-automation systems by providing customized assistance in real time to an individual user based on his skill-level, which could reduce interference from the automation when unnecessary, thereby enhancing the effectiveness of the shared control. The proposed hybrid shared controller was modeled as a hybrid system whose discrete mode denotes the control authority allocation. Then, an optimal control authority allocation is chosen by minimizing a loss function which represents the expected difference between the expert's and a novice's performance. We tested and demonstrated the efficacy of the proposed scheme with an illustrative example using a quadrotor landing simulator.

ACKNOWLEDGMENT

The Institutional Review Board at Purdue University approved the study.

REFERENCES

[1] S. M. Erlien, S. Fujita, and J. C. Gerdes, "Shared steering control using safe envelopes for obstacle avoidance and vehicle stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 441–451, 2016.

[2] N. Amirshirzad, A. Kumru, and E. Oztop, "Human adaptation to humanrobot shared control," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, pp. 126–136, 2019.

[3] B. Khademian and K. Hashtrudi-Zaad, "Dual-user teleoperation systems: New multilateral shared control architecture and kinesthetic performance measures," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 5, pp. 895–906, 2011.

[4] G. Aguirre-Ollinger, J. E. Colgate, M. A. Peshkin, and A. Goswami, "Active-impedance control of a lower-limb assistive exoskeleton," in *2007 IEEE 10th international conference on rehabilitation robotics*. IEEE, 2007, pp. 188–195.

[5] M. Cubuktepe, N. Jansen, M. Alshiekh, and U. Topcu, "Synthesis of provably correct autonomy protocols for shared control," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[6] D. A. Abbink, T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systemsfinding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[7] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: Smoothly shifting control authority?" *Cogn. Technol. Work*, vol. 14, no. 1, p. 1928, Mar. 2012.

[8] S. Musi and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342–354, 2017.

[9] M. Flad, L. Fröhlich, and S. Hohmann, "Cooperative shared control driver assistance systems based on motion primitives and differential games," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 711–722, 2017.

[10] S. J. Anderson, S. B. Karumanchi, K. Iagnemma, and J. M. Walker, "The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 45–54, 2013.

[11] A. Bhardwaj, Y. Lu, S. Pan, N. Sarter, and B. Gillespie, "The effects of driver coupling and automation impedance on emergency steering interventions," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 1738–1744.

[12] A. T. Nguyen, J. J. Rath, C. Lv, and T. M. Guerra, "Human-machine shared control for semi-autonomous vehicles using level of cooperativeness," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2770–2775.

[13] B. Pano, F. Claveau, P. Chevrel, C. Sentouh, and F. Mars, "Systematic h2/h-inf haptic shared control synthesis for cars, parameterized by sharing level," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 4416–4423.

[14] A. Broad, I. Abraham, T. Murphey, and B. Argall, "Data-driven koopman operators for model-based shared control of humanmachine systems," *The International Journal of Robotics Research*, vol. 39, no. 9, p. 11781195, Jun 2020.

[15] J. Jiang and A. Astolfi, "State and output-feedback shared-control for a class of linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3209–3214, 2016.

[16] C. Alexander Braun, C. Bohn, J. Inga, and S. Hohmann, "A cooperative assistant system with smoothly shifting control authority based on partially observable markov decision processes," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 806–811.

[17] H. Konishi, N. Sakagami, T. Wada, and S. Kawamura, "Haptic shared control for path tracking tasks of underwater vehicles," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 4424–4430.

[18] K. Fitzsimons, A. Kalinowska, J. P. Dewald, and T. D. Murphey, "Task-based hybrid shared control for training through forceful interaction," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1138–1154, 2020.

[19] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 475–491, 2020.

[20] Z. Ercan, A. Carvalho, M. Gokasan, and F. Borrelli, "Modeling, identification, and predictive control of a driver steering assistance system," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 700–710, 2017.

[21] K. Blanchet, A. Bouzeghoub, S. Kchir, and O. Lebec, "How to guide humans towards skills improvement in physical human-robot collaboration using reinforcement learning?" in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 4281–4287.

[22] W. Jin, D. Kulić, J. F.-S. Lin, S. Mou, and S. Hirche, "Inverse optimal control for multiphase cost functions," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1387–1398, 2019.

[23] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "Online dynamic mode decomposition for time-varying systems," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 3, pp. 1586–1609, 2019.

[24] D. Liberzon, *Switching in Systems and Control*. Springer, 2003.

[25] Y. Zhao, B. Pano, P. Chevrel, F. Claveau, and F. Mars, "Driver model validation through interaction with varying levels of haptic guidance," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2284–2290.

[26] F. Sabatino, "Quadrotor control: modeling, nonlinearcontrol design, and simulation," Master's thesis, KTH, Automatic Control, 2015.